

Calculateur de Complément

En Langage C

Code source:

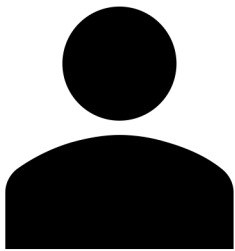
<https://github.com/FaroukBel/Ones-Twos-Complement>

FAROUK BEL KHYATE

AMINE ASSALANE

Présentation Du Problème

Utilisateur



Nombre Décimale: -C

Complement.exe

Valeur signée en binaire: 10001100

Complément a un: 11110011

Complément a deux: 11110100

Algorithmes

Complément

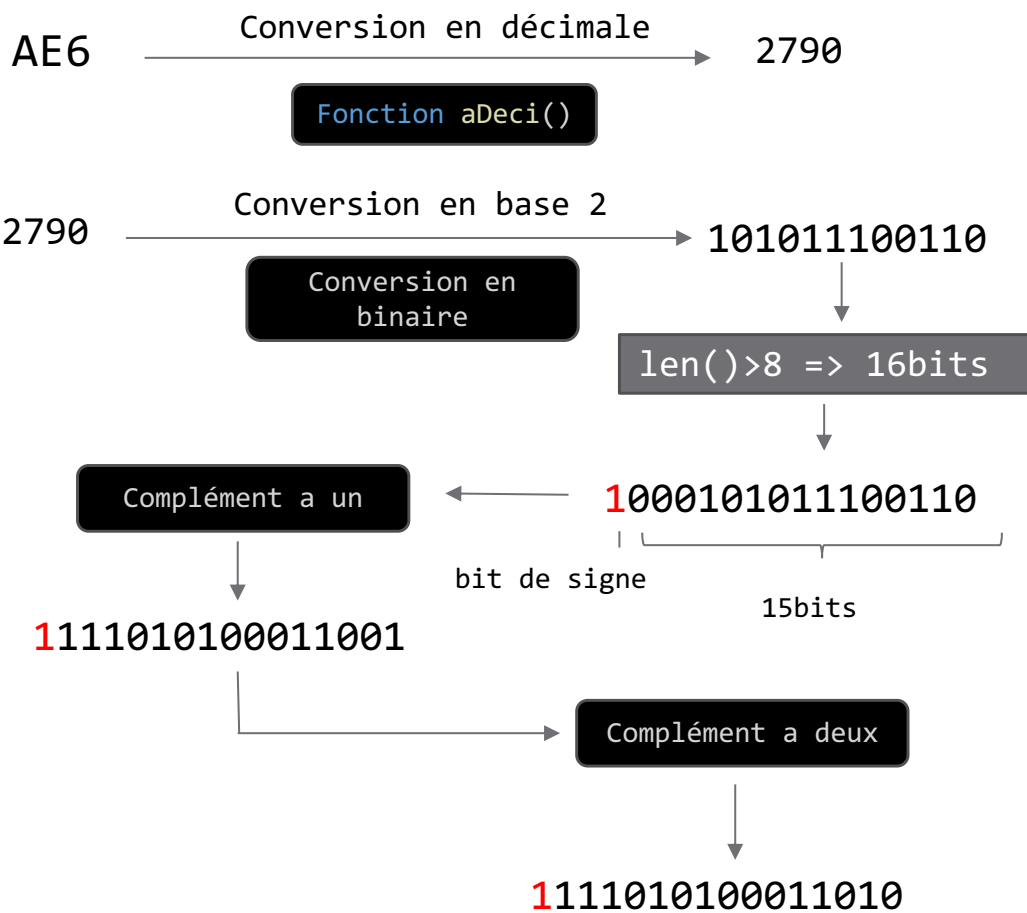
Utilisateur



Nombre: -AE6
Base d'origine: 16

Fonction main()

Complement.exe



Fonction de saisie et d'exécution des algorithmes

```
int main()
{

int n, entier, base1;
char str[100];

printf("Entrer un nombre: ");
scanf("%s", str);
printf("Entrer sa base: ");
scanf("%d", &base1);
printf("\n");
```

L'utilisateur saisie le nombre dans un tableau str pour l'utiliser comme un string et traiter chaque case

```
// Condition pour le nombre 0 et -0
if ((str[0] == '-' && str[1] == '0') || str[0] == '0')
{
printf("Les differentes representations de -0 :");
printf("La valeur signee: 00\n");
printf("Le complement a 1: 11\n");
printf("Le complement a 2: 00\n\n");
printf("Les differentes representations de 0 :");
printf("La valeur signee: 00\n");
printf("Le complement a 1: 00\n");
printf("Le complement a 2: 00\n");
}
```

Cette condition traite l'exception du nombre 0 et -0

```
// Condition de traitement des nombres negatifs
else if (str[0] == '-')
{
    // Boucle pour traite juste le nombre sans signe
    (car c'est un string)
    for (int i = 0; i < strlen(str); i++)
    {
        str[i] = str[i + 1];
    }
}
```

Dans cette condition on traite les valeurs negatives et la boucle enlève le signe pour travailler juste sur le nombre donné par l'utilisateur

```
entier = aDeci(str, base1);
n = entier;
int j = len(entier);

int T[j];
int i = 0, l = 0;

// Conversion binaire
for (i = 0; i < j; i++)
{
    T[i] = n % 2;
    n = n / 2;
}
```

Après on convertit le nombre donné en base décimale à l'aide de la fonction `aDeci()` puis en base binaire

Fonction

Complément

```
// Inverser le tableau
for (i = j - 1; i >= 0; i--)
{
    vs[i] = T[l];
    l++;
}

// Ajoutant le 1 de signe en l'indice 0
vs[0] = 1;

printf("La valeur signee: ");

for (i = 0; i < j; i++)
    printf("%d", vs[i]);

printf("\n");
```

On inverse le tableau precedant pour completer la conversion en binaire et on ajout 1 au premier indice pour indiquer la valeur signee

```
int ca1[j + 1];
ca1[0] = 1;

// Complement a un on inversant 1 par 0 et 0 par 1 pour
chaque
// case du tableau vs[]

for (i = 1; i < j + 1; i++)
{
    if (vs[i] == 0)
    {
        ca1[i] = 1 / vs[i] + 1;
    }

    else
    {
        ca1[i] = vs[i] - 1;
    }
}
```

Fonction

Complément

```
printf("Le complement a 1: ");  
  
for (i = 0; i < j; i++)  
    printf("%d", ca1[i]);  
printf("\n");
```

Le complement a 1 se fait on inversons les elements du tableau precedent (1 par 0 et vise versa) et le 1 de signe dans l'indice 0

```
int ca2[j];  
int reste = 1; // Inisiation du reste a un  
  
// Complement a deux inversant 1 par 0 si le reste est  
// a 1 et  
// on inverse 0 par 1 si l'indice est a 0 et le reste  
// est a 1  
// et le reste dans ce cas devient 0 pour que l'indice  
// suivant  
// soit le meme  
for (int i = j - 1; i >= 0; i--)  
{  
    if (ca1[i] == 1 && reste == 1)  
    {  
        ca2[i] = 0;  
    }  
    else if (ca1[i] == 0 && reste == 1)  
    {  
        ca2[i] = 1;  
        reste = 0;  
    }  
    else  
    {  
        ca2[i] = ca1[i];  
    }  
}  
ca2[j] = '\\0';
```

Fonction

Complément

```
printf("Le complement a 2: ");  
  
for (i = 0; i < j; i++)  
    printf("%d", ca2[i]);  
printf("\n");
```

Le complement a 2 se fait on inversons les elements du tableau precedent (1 par 0 et vise versa) et le 1 de signe dans l'indice 0

```
// Traitement des nombres positifs  
else if (str[0] != '-')  
{  
  
    entier = aDeci(str, base1);  
    int j = len(entier);  
    int T[j];  
    int i = 0, l = 0;  
    // Conversion binaire  
    for (i = 0; i < j; i++)  
    {  
        T[i] = entier % 2;  
        entier = entier / 2;  
    }  
  
    int vs[j];  
  
    // Inverser le tableau  
    for (i = j - 1; i >= 0; i--)  
    {  
        vs[i] = T[l];  
        l++;  
    }  
}
```

La procedure des nombres positif est la meme que les nombres negatifs sauf que la valeur signee et le complement a un et a deux sont les memes


```
// Conversion ASCII d'un nombre en hexa a un nombre entier
int val(char c)
{
    if (c >= '0' && c <= '9') // Verification par code ASCII
        return (int)c - '0'; // Cast du char en int
    else
        return (int)c - 'A' + 10;
}
```

Fonction val convertir un char a entier

```
// Fonction pour convertir un nombre en decimale
int aDeci(char *str, int base1)
{
    int len = strlen(str);
    int power = 1;
    int num = 0;
    int i;

    // Equivalent de decimale = str[len-1]*1 + str[len-2]*base + str[len-3]*(base^2) + ...
    for (i = len - 1; i >= 0; i--)
    {
        num += val(str[i]) * power;
        power = power * base1;
    }

    return num;
}
```

Cette fonction convertir de telle base au base decimale

```
// Fonction pour détecter le nombre de bits optimale
int len(int entier)
{
    int j = 0;
    while (entier > 0)
    {
        j = j + 1;
        entier = entier / 2;
    }

    if (j < 4 && j >= 1)
    {
        j = 4;
    }
    else if (j >= 4 && j < 8)
    {
        j = 8;
    }
    else if (j >= 8 && j <= 16)
    {
        j = 16;
    }
    else
    {
        j = 32;
    }
    return j;
}
```

Fonction val convertir un char a entier
on utilisons code ASCII

Execution

Complement

```
Ones-Twos-Complement — -zsh — 80x24
[faroukbelkhyate@192 Ones-Twos-Complement % ./main
Entrez un nombre: AE6
Entrez son base: 16

La valeur signee: 0000101011100110
Le complement a 1: 0000101011100110
Le complement a 2: 0000101011100110
[faroukbelkhyate@192 Ones-Twos-Complement % ./main
Entrez un nombre: -AE6
Entrez son base: 16

La valeur signee: 1000101011100110
Le complement a 1: 1111010100011001
Le complement a 2: 1111010100011010
faroukbelkhyate@192 Ones-Twos-Complement %
```

Execution pour tester un nombre en hexadecimale

```
Ones-Twos-Complement — -zsh — 80x24
[faroukbelkhyate@192 Ones-Twos-Complement % ./main
Entrez un nombre: -ZS
Entrez son base: 36

La valeur signee: 1000010100001000
Le complement a 1: 1111101011110111
Le complement a 2: 1111101011111000
faroukbelkhyate@192 Ones-Twos-Complement %
```

Execution pour tester un nombre en base 36

Aventages:

- Ce program peut faire le calcule des complement un et deux d'un nombre de base 2 -> 36
- Convertir d'une base quelconque au decimale

Limitations:

- Ce program ne peut pas faire une conversion des nombres flottants
- Il faut toujours enter un nombre en majuscule (cas d'un nombre qui contient des lettres)